no. 442-70 CANCELLED

[illegible] THE GENERAL

n/m JOB-SHOP PROBLEM

by

[illegible] Fisher

[illegible] 1970

[illegible] Paper [illegible]

GROUP THEORY IN THE GENERAL n/m JOB-SHOP PROBLEM

by

Marshall L. Fisher

February 9, 1970

Working Paper 443-70

ABSTRACT

The n/m Job-Shop Problem is formulated as an Integer Program and the asymptotic algorithm of integer programming is applied. A simple way of getting the group problem is obtained and some results of the structure of the group problem are derived. Two problems are found to arise with this approach: (1) the group problem is larger than can be handled by current shortest route algorithms, and (2) the relaxed non-negativity constraints will be violated by at least one optimal group problem solution. Future research will attempt to resolve these problems.

Group Theory in the General n/m Job-Shop Problem

Introduction

Conway, Maxwell and Miller have summarized past research on the general n/m job-shop problem with these words of encouragement:

> "The general job-shop problem is a fascinating challenge. Although it is easy to state, and to visualize what is required, it is extremely difficult to make any progress whatever toward a solution. Many proficient people have considered the problem, and all have come away essentially emtpy-handed. Since this frustration is not reported in the literature, the problem continues to attract investigators..."[1]

In this paper, the group theoretic approach to integer programming[2] will be applied to this problem. We will formulate this problem as an Integer Program and derive a group problem from the optimal LP solution ot this formulation. This problem is characterized by large size and considerable structure. For example, the 5/3 problem given in Figure 1 yields a group of size $(40)^{11}$. However, the structure of this group seems very simple; it can be represented by 11 equations involving 24 variables whose coefficients are all 0, 1 or -1. The relevant question will be how to use the structure of the problem to allow treatment of the large groups. A second problem which must also be handled is that solutions to the group

---

[1] Conway, R.W., W.L. Maxwell, and L.W. Miller, Theory of Scheduling, Addison-Wesley, 1967, p. 103.

[2] Gomory, R.E., "On the Relation Between Integer and Non-Integer Solutions to Linear Programs," Proceeding of the National Academy of Sciences, 53, pp. 250-256, (1965).

Shapiro, J., "Dynamic Programming Algorithms for the Integer Programming Problem--I: The Integer Programming Problem Viewed as a Knapsack Type Problem," Operations Research 16, pp. 103-121, (1968), and "Group Theoretic Algorithms for the Integer Programming Problem II: Extension to a General Algorithm," Operations Research 16, pp. 928-947, (1968).

In section 1 of this paper an integer programming formulation of the Job Shop Scheduling problem is presented. We then relax the integrality requirements in this formulation to obtain an LP problem. In section 2 we show that the solution of this LP problem is easily obtained by Critical Path Methods and discuss which variables in this solution will be non-integer. In section 3 we derive a Group Problem which characterizes solutions in which these non-integer variables are made integer. In section 4 we discuss the structure of this Group Problem. Section 5 illustrates our results with a numerical example and section 6 contains some concluding remarks.

1. Integer Programming Formulation

The n/m job-shop problem involves n jobs which must each be processed on m different machines. The sequence in which a given job is processed on each machine is fixed and not necessarily the same for 2 different jobs. For example, job 1 might be processed first on machine 1, then 2, then 3, etc., while job 2 is processed first on machine 3, then 1, then 2, etc.

The processing of a job on a machine is called a task. Tasks have predetermined process times (possibly zero) and for a given job, all preceding tasks must be complete before a particular task can be started. The problem is to determine a sequence for processing each job on each machine so that some criterion is optimized. Here we'll restrict our attention to minimizing make-span, the time to complete all jobs.

A given problem can be represented by an m x n matrix of task process times and another m x n matrix showing the processing sequence for each job. This data is depicted in Figure 1 for a 5/3 problem. In the tables

shown, the jobs are numbered 1 to 5 and tasks 1 to 3 with 1 being the first task in time performed on a job. Machines are labeled A, B, and C. The second table relates task sequence to machines by giving the machine on which task 1 of job 1 is performed, etc. The data for this problem was generated randomly and will be used throughout the paper for illustration.

Figure 1

Job/Task	1	2	3
1	9	3	8
2	3	3	0
3	6	7	5
4	5	5	5
5	2	7	0

Process Times

Job/Task	1	2	3
1	B	C	A
2	A	C	B
3	B	A	C
4	B	A	C
5	B	C	A

Sequence

There have been several linear integer programming formulations of this problem; the most recent and compact one given here is due to Wagner.[3] We let t_{ij} be a non-negative variable giving the start time of the j^{th} task on the i^{th} job and p_{ij} be a constant process time for the same task. (p_{ij} will be an element of the first matrix in Figure 1.) Even though t_{ij} is not required to be integer from its physical interpretation as a time, we will see in sections 2 and 3 that because of the problem structure, if p_{ij} is integer, t_{ij} will always be integer in an optimal solution. This allows us to treat t_{ij} as though it were required to be integer and use all-integer

[3]Wagner, H.M., "An Integer Linear-Programming Model for Machine Scheduling," NRLQ, 6, No. 2, June 1959.

programming techniques. Finally, we let $y_{i\ell}^{k}$ be a 0-1 variable equal to 1 if job i preceeds job ℓ on machine k or 0 if ℓ preceeds i and M a sufficiently large positive integer, and the formulation preceeds as follows.

First, there are task precedence contraints for each job.

(1) $t_{ij} \geq t_{i,j-1} + p_{i,j-1}$ $\quad$ j+2,...m, i = 1,...n

Then, constraints which say, in effect, only one job can use a machine at once.

$t_{ij} - t_{\ell p} \geq p_{\ell p} - y_{i\ell}^{k} \cdot M$ $\quad$ k = 1,...,m
all ij & ℓp for which both tasks are processed on machine k.

(2) $t_{\ell p} - t_{ij} \geq p_{ij} - (1-y_{i\ell}^{k}) \cdot M$

In these constraints, the 0-1 variable and the large positive M are used in the familiar way of representing conjunctive constraints to relax exactly one of the constraints depending on which job is first on machine k (and hence on what the value of $y_{i\ell}^{k}$ is).

M must satisfy the following relations, where T* and T_* are the maximum and minimum values assumed by $(t_{\ell p} - t_{ij})$ in any allowable optimal solution.

(2b) $M \geq T^* + P_{\ell p}$

$M \geq -T_* + P_{ij}$

To determine T* and T_* we must have an upper bound on the objective equation from which it is easy to deduce an allowable range of optimal values for $(t_{\ell p} - t_{ij})$. Note that although we have assumed one constant M for

notational ease, in practice we could actually use a different value in each equation of (2).

Finally, the objective can be represented with the following constraints:

$$\min z = t_f$$

(3)

$$t_f \geq t_{im} + p_{im} \qquad i = 1,\ldots,n$$

In all, there will be $(m - 1)\cdot n$ equations of type (1), $(n - 1)n\cdot m$ of type (2), and n of type (3). There will be $\frac{(n-1)\cdot n}{2}\cdot m$ 0-1 variables whose values represent a particular sequence of jobs on each machine and $n\cdot m$ non-negative t_{ij} variables giving the start time of each task.

2. LP Solutions of the Problem

In this section we will show what form the optimal LP solution to this problem will take.

Allowing the 0-1 variables $y^k_{i\ell}$ to assume non-integer values is equivalent to relaxing (2). In this case there is a simple optimal solution in which

$$t_{i1} = 0$$

(4)

$$t_{ij} = t_{ij-1} + p_{ij-1} \qquad j = 2,\ldots,m,\ i = 1,\ldots,n$$

$$t_f = \max t_{in}$$

All slacks in (1) will be zero, and one slack in (3) will be zero, the rest basic. Figure 2 shows a network interpretation of this solution for our 5/3 example, while Figure 3 depicts the machine usage implied by this solution. Note that the solution is clearly not feasible for the

scheduling problem since some tasks overlap on the times they use machines. There are a number of alternate optimum values for the 0-1 $y_{i\ell}^{k}$ variables in (2) and for the slacks associated with those variables. Consider the general form of one of these equations as:

$$
\begin{aligned}
t_1 &= s_1 + t_2 + p_2 - y \cdot M \\
t_2 &= s_2 + t_1 + p_1 - (1-y) \cdot M
\end{aligned}
\tag{5}
$$

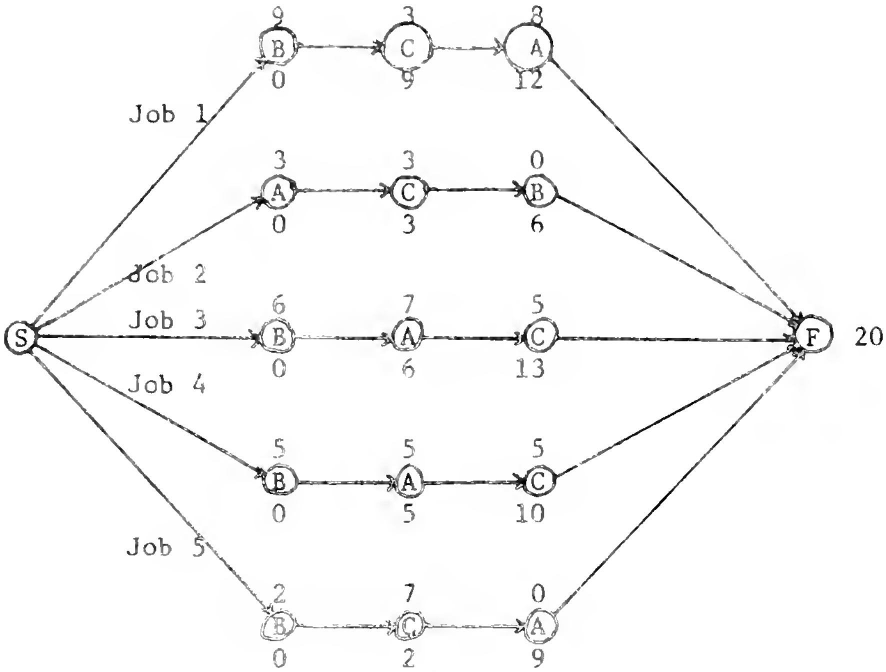
where we have added slacks s_1 and s_2 and simplified notation for this discussion by dropping subscripts on $y_{i\ell}^{k}$ and changing subscripts on t_{ij} and $t_{\ell p}$. t_1 and t_2 are task start times whose values are assumed to be predetermined by (4), p_1 and p_2 are constant process times, and M is a constant as previously defined. Since the slacks s_1 and s_2 and the 0-1 variable y appear only in the two equations shown here, we can set values for them without worrying about the rest of the problem equations. Our objective is to find basic values to these variables which satisfy (5) (and hence (2)). Along with the values set by (4), these will form a basic, optimal LP solution.

One possible set of values is:

$$
\begin{aligned}
y &= \frac{M + t_2 - t_1 - p_1}{M} \\
s_1 &= M - p_1 - p_2 \\
s_2 &= 0
\end{aligned}
\tag{6}
$$

If M is set according to (2b) then y will be positive, one of the slacks will be zero and the other non-negative so this solution is

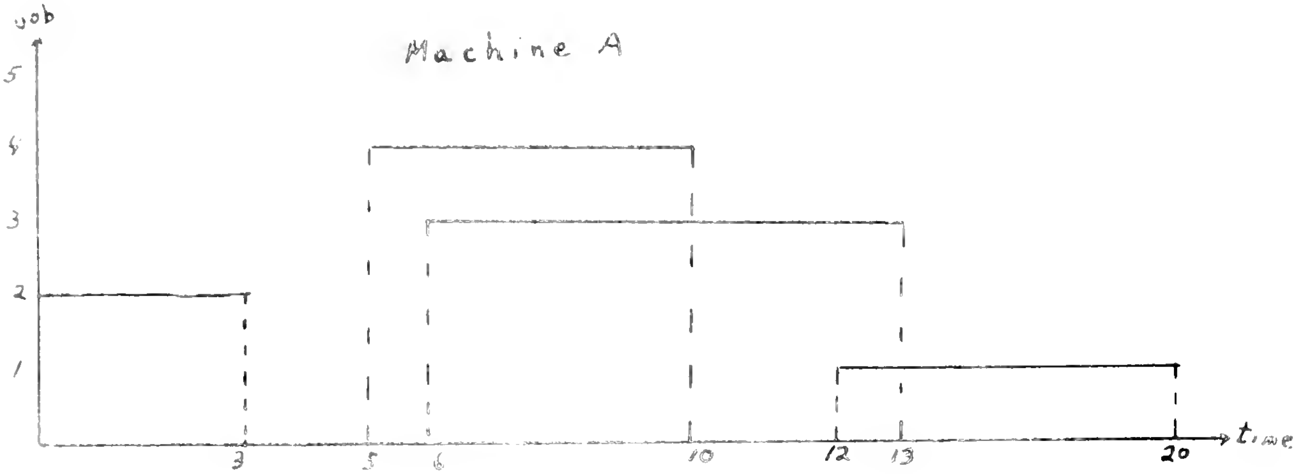
Figure 2

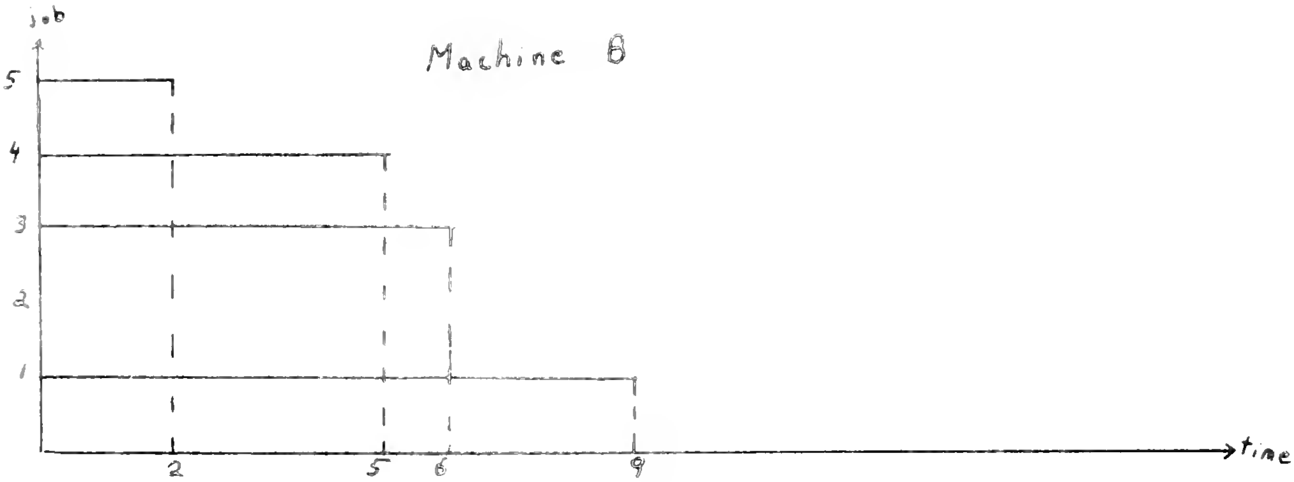


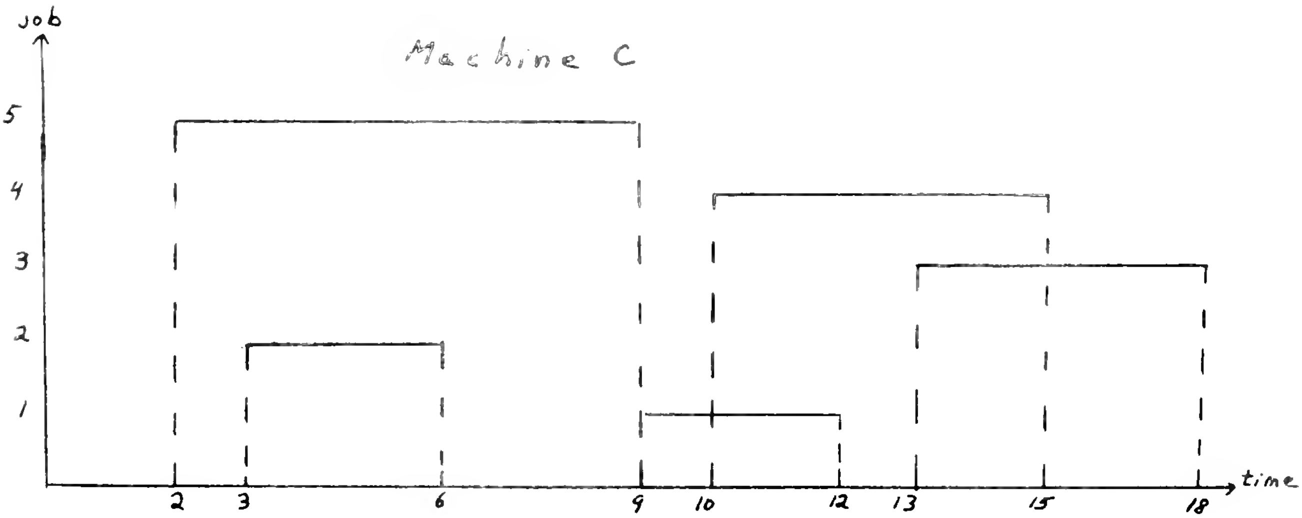
Graph for the Optimal LP Solution of the 5/3 Example

Nodes correspond to tasks. The number over a task is the process time, the number below the start time. The number beside F is the value of the objective function.

Figure 3 Gantt Charts for the LP Solutions to the 5/3 Problem







basic and feasible.

Note that this set of values implies that in general all of the 0-1 variables will have positive fractional values. There is an alternate set of optimal values in which many of the 0-1 variables can be made nonbasic, thus reducing the size of the group problem obtained. These values are:

$$y = \begin{cases} 0 & \text{, if } t_1 \geq t_2 + p_2 \\ 1 & \text{, if } t_2 \geq t_1 + p_1 \\ \dfrac{M+t_2-t_1-p_1}{M} & \text{, otherwise} \end{cases}$$

$$s_1 = \begin{cases} t_1-t_2-p_2 & \text{, if } t_1 \geq t_2 + p_2 \\ t_1-t_2-p_2+M & \text{, if } t_2 \geq t_1 + p_1 \\ M-p_1-p_2 & \text{, otherwise} \end{cases}$$

$$s_2 = \begin{cases} t_2-t_1-p_1+M & \text{, } t_1 \geq t_2 + p_2 \\ t_2-t_1-p_1 & \text{, } t_2 - t_1 + p_1 \\ 0 & \text{, otherwise} \end{cases}$$

It will always be possible to redefine y so that it becomes 0 in the case when it would be 1 and we will assume that this is done. It can be seen that these values are also basic and feasible.

It is easy to see what we are doing by looking at Figure 3 for our 5/3 example. There will be one 0-1 variable and 2 equations like (2) for each possible pair of tasks on a machine, so in this example there are 22 0-1 variables. The values in (7) correspond to making y 0 or 1 whenever the tasks in Figure 3 do not overlap on a machine, and setting y as in (6) only when tasks overlap. Since there are only 11 overlaps

in our example, we have reduced the number of basic $y^k_{i\ell}$'s by half.

As we shall see, this corresponds to reducing the size of the group from M^{22} to M^{11}. However, we do loose something by this reduction, since we will end up dropping the equations of type (2) corresponding to non-basic $y^k_{i\ell}$'s as part of dropping non-negativity on basics and hence the solution to the group problem might violate some of these equations, i.e. produce overlap in tasks that previously didn't overlap. In this event, we would have to explore other solutions of the group problem to find one which is feasible in all equations. Nevertheless, it seems reasonable to expect that this will occur infrequently and that these equations are likely things to give up in order to reduce the size of the group problem. Therefore, we will assume in subsequent work that the form given in (7) is to be used with $y^k_{i\ell}$'s redefined to make those with the value 1 nonbasic. We might push this idea still further by attempting to juggle task start times to get a solution with fewer overlaps and no larger objective value, but we won't bother at this time.

3. Derivation of the Group Problem

In this section we will follow the strategy suggested by the group theoretic approach to integer programming.[4] Beginning with the optimal LP solution to (1), (2) and (3) which has some variables at non-integer values, we derive a group problem whose solution set corresponds to the set of "corrections" to (1), (2) and (3). A correction is a set of non-negative integer values for non-basic variables which force all basic

[4]Gomory, op. cit. Shapiro, op. cit.

variables to integer values. In determining a correction via the group problem we ignore the non-negativity constraints on the basic variables so the solution determined may still be infeasible. The group problem is determined by mapping all non-basic columns and the right hand side of the optimal LP tableau into their fractional parts.

We start by writing (1), (2), and (3) without the objective function in partitioned matrix form as follows:

$$
\begin{array}{llllllllll}
 & \text{(I)} & B_{11}T_B + R_{11}T_R & & -IS_{1R} & & & & & = b_1 \\
 & \text{(II)} & B_{21}T_B + R_{21}T_R & + B_{22}T_F & & + R_{22}S_{2R} & & & & = b_2 \\
(8) & \text{(III)} & B_{31}T_B + R_{31}T_R & & & & + A_3Y_B & & -IS_{3R} & = b_3 \\
 & \text{(IV)} & B_{41}T_B + R_{41}T_R & & & & + A_4Y_B & & -IS_{4B} & = b_4 \\
 & \text{(V)} & B_{51}T_B + R_{51}T_R & & & & & + A_5Y_R & & = b_5
\end{array}
$$

Where (I) represents equations (1), (II) equations (3) without the objective, (III) equations (2) in which a $y^k_{i\ell}$ is basic, (IV) the companions to the equations (2) represented in (III) in which slacks are basic, and V equations (2) in which no $y^k_{i\ell}$'s are basic. T_B and T_R are vectors of the basic and nonbasic task start time variables T_{ij}, T_F a vector including the variable t_f defined in (3) and the n-1 basic slacks associated with (3), Y_B and Y_R are the basic and non-basic 0-1 $y^k_{i\ell}$ variables, and S_{iB}, i = 1,2,3, and S_{iR}, i = 4,5, represent basic and non-basic slacks.

The entries of matrices B_{i1} and R_{i1}, i = 1,...,5, B_{22}, and R_{22} are all 0, 1 or -1. The entries of A_3, A_4, and A_5 are all M, -M, and 0.

The entries of b_i, $i = 1,\ldots,5$ will be process times or process times $\pm$ M and hence are all integers.

By the way we define y in (7) and by (2) we have $A_3 = (M)\cdot I$ and $A_4 = (-M)\cdot I$ where $\cdot$ indicates scalar multiplication. Also, note that the coefficients of t_{ij} and $t_{\ell p}$ in once equation of (2) are just the negative of the coefficients in the companion equation. Thus, if we associate corresponding rows of 8(III) and 8(IV) with the same machine equations we will have $B_{31} = -B_{41}$ and $R_{31} = -R_{41}$. Moreover, assuming $m \geq 2$ by (3) and (4) we have $R_{21} = 0$.

Using these facts the optimal LP basis for (8) can be written as

$$B = \begin{bmatrix} B_{11} & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 & 0 \\ B_{31} & 0 & (M)\cdot I & 0 & 0 \\ -B_{31} & 0 & (-M)\cdot I & -I & 0 \\ B_{51} & 0 & 0 & 0 & -I \end{bmatrix}$$

and it is easy to check that its inverse is

$$B^{-1} = \begin{bmatrix} B_{11}^{-1} & 0 & 0 & 0 \\ -B_{22}^{-1}B_{21}B_{11}^{-1} & B_{22}^{-1} & 0 & 0 \\ (-\frac{1}{M})\cdot B_{31}B_{11}^{-1} & 0 & (\frac{1}{M})\cdot I & 0 \\ 0 & 0 & -I & 0 \\ -B_{51}B_{11}^{-1} & 0 & 0 & -I \end{bmatrix}$$

Multiplying (8) by B^{-1} yields:

$$
(9)\quad
\begin{array}{llllll}
\text{(I)} & IT_B + B_{i1}^{-1}R_{11}T_R & - B_{11}^{-1}S_{1R} & & & = B_{11}^{-1}b_1 \\
\text{(II)} & -B_{22}^{-1}B_{21}B_{11}^{-1}R_{11}T_R & + IT_F + B_{22}^{-1}B_{21}B_{11}^{-1}S_{1R} + B_{22}^{-1}R_{22}S_{2R} & & & = -B_{22}^{-1}B_{21}B_{11}^{-1}b_1 + B_{22}^{-1}b_2 \\
\text{(III)} & (-\frac{1}{M})\cdot(B_{31}B_{11}^{-1}R_{11}-R_{31})T_R & (\frac{1}{M})\cdot B_{31}B_{11}^{-1}S_{1R} & + IY_B & - (\frac{1}{M})\cdot IS_{3R} & = (-\frac{1}{M})\cdot(B_{31}B_{11}b_1-b_3) \\
\text{(IV)} & & & & + IS_{3R} + IS_{4B} & = -b_3-b_4 \\
\text{(V)} & (B_{51}B_{11}^{-1}R_{11}-R_{51})T_R & B_{51}B_{11}^{-1}S_{1R} & - A_5Y_R & + IS_{5B} & = B_{51}B_{11}^{-1}b_1-b_5
\end{array}
$$

We will now show that B_{11}^{-1} and B_{22}^{-1} are integer matrices. From this it can be seen that all entries in (9)(I), (II), (IV), (V) are integer so that when we take their fractional parts in deriving the group problem we will get zeros. Thus they can be dropped from consideration leaving (9) (III) from which the group problem will be obtained.

First note from (1) that if we write the equations defined by (1) in the order $i = 1$, $j = 2,\ldots,m$, $i = 2$, $j = 2,\ldots,m,\ldots i = n$, $j = 2,\ldots,m$ and order the variables t_{ij} as $t_{11},\ldots,t_{1n}$, $t_{21},\ldots,t_{nm}$ then B_{11} will have the form

$$B_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots\cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots\cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots\cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots\cdots & 0 \\ 0 & 0 & 0 & -1 & \cdots\cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots\cdots & 1 \end{bmatrix}$$

which is an identity matrix with a few -1's just below the diagonal. More precisely,

$$B_{11} = I - E$$

where $E = ||e_{ij}||$ and:

$$(10) \qquad e_{ij} = \begin{cases} 0 \text{ or } 1 & j = i-1,\ i = 2,\ldots,n \\ 0 & \text{otherwise} \end{cases}$$

Lemma 1: Let E be an nxn matrix as defined by (10). Then $E^n = 0$.

Proof: Let $E^2 = ||e^2_{ij}||$. Then, using (10)

$$e^2_{ij} = \sum_{K=1}^{n} e_{iK} e_{jK} = e_{i,i-1} e_{i-1,i-2} \qquad i = 3,\ldots,n$$

so

$$e^2_{ij} = \begin{cases} 1 & j = i-2,\ i = 3,\ldots,n \\ 0 & \text{otherwise} \end{cases}$$

Now, using induction, assume $E^\ell = ||e^\ell_{ij}||$ has the form

(11) $$e^\ell_{ij} = \begin{cases} 1 & j = i-\ell,\ i = 1+\ell,\ldots,n \\ 0 & \text{otherwise} \end{cases}$$

and show $e^{\ell+1}_{ij}$ has the same form.

$$e^{\ell+1}_{ij} = \sum_{K=1}^{n} e_{iK} e^{\ell+1}_{Kj} = e_{i,i-1} e^{\ell}_{i-1,i-\ell-1} = \begin{cases} 1 & j = i-\ell+1 \\ & i = 2+\ell,\ldots,n. \\ 0 & \text{otherwise} \end{cases}$$

Hence E^n has form (11) and equals 0.

Lemma 2: $B^{-1}_{11} = (I + E + E^2 + \ldots E^{n-1})$

Proof: $$B_{11}B^{-1}_{11} = (I - E)(I + E + E^2 + \ldots E^{n-1})$$

$$= I - E + E - E^2 + E^2 + \ldots -E^{n-1} + E^{n-1} - E^n$$

$$= I - E^n$$

$$= I.$$

Since E is integer, E^j is integer and B^{-1}_{11} is integer.

Using (3) we can see that B_{22} will have the form

$$B_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & -1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & -1 \end{bmatrix}$$

where the first column corresponds to t_f, the first row corresponds to the single binding equation, and subsequent columns correspond to basic slacks.

Lemma 3: $B_{22}^2 = I$

Proof: Let $B_{22}^2 = ||b_{ij}^2||$ and $B_{22} = ||b_{ij}||$. Then

$$b_{1j}^2 = \sum_{K=1}^{n} b_{1K}b_{Kj} = b_{11}b_{1j} = \begin{cases} 1 & j = 1 \\ 0 & j \neq 1 \end{cases}$$

$$\underset{i\neq 1}{b_{ij}^2} = \sum_{K=1}^{n} b_{iK}b_{Kj}$$

$$= b_{i1}b_{1j} + b_{ii}b_{ij}$$

$$= \begin{cases} 1 - 1 & j = 1 \\ 0 + 1 & j = i, \quad i \neq 1 \\ 0 + 0 & j \neq i,1 \end{cases}$$

Again, since B_{22} is all integer, B_{22}^{2} will be also. Thus B_{11}^{-1} and B_{22}^{-1} are integer and it can easily be seen that all matrices and right hand sides in equations (9) I, II, IV, V are integer and they can be dropped from the group problem, allowing us to concentrate on (9) III for the group equations.

Writing III as a group problem, after multiplying through by M we get

$$(12) \qquad (-B_{31}B_{11}^{-1}R_{11}+R_{31})T_R+B_{31}B_{11}^{-1}S_{1R} - IS_{3R} \equiv -B_{31}B_{11}^{-1}b_1 + b_3 \quad (\text{mod } \underline{M\cdot 1})$$

where (mod $\underline{M\cdot 1}$) means each row is taken modulo M, and T_R, S_{1R}, and S_{3R} are integer.

4. Structure of the Group Problem

In this section we will look in greater detail at the structure of the group problem expressed in (12) and also derive the linear objective function for the group problem.

A single equation from (8)(III) may be written as

$$(13) \qquad t_{\ell p} - t_{ij} - S_{i\ell}^{k} - My_{i\ell}^{k} = p_{ij} - M$$

where the j^{th} task on job i and p^{th} task on ℓ both use machine k and overlap in the current LP solution. $S_{i\ell}^{k}$ is the slack on (2).

In the optimal LP tableau (9)(III), (13) will be basic relative to $y_{i\ell}^{k}$. Hence to put (13) in optimal tableau form we must eliminate all other basic variables. $S_{i\ell}^{k}$ is non-basic by the way we constructed our optimal LP solution in (7). If j = 1, then $t_{ij} = t_{i1} = 0$ by (4) and is non-basic. Otherwise, in general $t_{ij} \neq 0$ and is basic.

In this case we write a subset of the precedence constraints for job i given in (1) as follows:

$$
\begin{array}{lllllll}
 & t_{i2} - t_{i1} & - S_{i1} & & & = p_{i1} \\
t_{i3} - t_{i2} & & & - S_{i2} & & = p_{i2} \\
 & \vdots & & & & \vdots \\
t_{ij} - t_{ij} - 1 & & & & - S_{i,j-1} & = p_{ij} - 1
\end{array}
$$

where we have introduced S_{ir}, $r = 1,\ldots,j-1$ as slacks. Adding these constraints together we get:

(14) $$t_{ij} - t_{i1} - \sum_{r=1}^{j-1} S_{ir} = \sum_{r=1}^{j-1} p_{ir}$$

Adding (14) and (13) yields

$$t_{pq} - t_{i1} - \sum_{r=1}^{j-1} S_{ir} - S^k_{i\ell} - My^k_{i\ell} = \sum_{r=1}^{j} p_{ir} - M$$

Now, if $q = 1$, t_{pq} is non-basic and we're done. If not, we can repeat the same process. For all cases we can write our result in the following form:

$$t_{\ell 1} + \sum_{r=1}^{p-1} S_{\ell r} - t_{i1} - \sum_{r=1}^{j-1} S_{ir} - S^k_{i\ell} - My^k_{i\ell} = \sum_{r=1}^{j} p_{ir} - \sum_{r=1}^{p-1} p_{\ell r} - M$$

where we interpret $\sum_{r=1}^{j-1}$ and $\sum_{r=1}^{p-1}$ as 0 if j or p = 1.

All variables in this equation are non-basic except y^k_{ip}. If we divide by M we get the optimal LP equation. Taking this mod 1 gives the

corresponding equation in the group problem. If we multiply by M and take the result mod M we get the following equivalent group equation:

$$(15)\qquad t_{\ell 1} + \sum_{r=1}^{p-1} S_{\ell r} - t_{i1} - \sum_{r=1}^{j-1} S_{ir} - S_{i\ell}^{k} = \sum_{r=1}^{j} p_{ir} - \sum_{r=1}^{p-1} p_{\ell r} \pmod{M}$$

If there are K overlaps of jobs on machines the group problem will consist of K of these equations and the size of the group will be M^k.

We will obtain the objective for the group problem is a similar manner. Following the group theoretic approach we want the LP objective expressed in terms of reduced costs and non-basic variables.

One of the constraints in (3) will be binding, say for i = b, and correspond to the longest job. In this equation t_f is basic.

$$t_f - S_{f1} - t_{bm} = p_{bn}$$

If we add (14) to this with i = b and j = n we get

$$t_f - S_{f1} - t_{b1} - \sum_{r=1}^{j-1} S_{br} = \sum_{r=1}^{n} p_{br}$$

or, for our group problem objective

$$(16)\qquad \min t_f = \sum_{r=1}^{n} p_{br} + t_{b1} + \sum_{r=1}^{j-1} S_{br}$$

where we dropped S_{f1} since it doesn't appear in (15). Note that this objective is tantamount to minimizing the completion time of the job which is longest in the LP solution. This could be a very poor objective to use if another job became longest in the process of obtaining an integerizing correction. This occurrence corresponds to one of the

basic slacks in (3) going negative and illustrates one of the problems with this approach.

(15) and (16) give us considerable insight into the structure of the group problem. Note, for example, that all coefficients will be 0, -1, or 1. Also, notice that we can always construct an optimal solution to the group problem using the slacks $S^k_{i\ell}$ which has a 0 objective value since these slacks don't appear in (16). This solution will always be infeasible in terms of non-negativity of the basic variables and illustrates another problem to be overcome with this approach.

We now have a constructive procedure for easily getting the group problem by inspection from a CPM solution and Gantt chart as shown in figures 2 and 3. This procedure is:

(1) Identify all job overlaps on machines.

(2) For each overlap, write an equation like (15).

(3) Identify the longest job and write an objective equation like (16).

5. Numerical Example

Using the procedure just outlined we can derive the group problem for the 5/3 example presented in figures 1, 2, and 3.

Step (1). Examining figure 3 the following 11 overlaps can be identified:

Machine A - Jobs 3 and 4, 3 and 1

Machine B - 5 and 4, 5 and 3, 5 and 1, 4 and 3, 4 and 1, 3 and 1

Machine C - 5 and 2, 1 and 4, 4 and 3

Step (2). The group equations for these overlaps are written below in matrix form.

Variables	t_{11}	t_{21}	t_{31}	t_{41}	t_{51}	S_{11}	S_{12}	S_{21}	S_{31}	S_{32}	S_{41}	S_{42}	S_{51}	S^A_{34}	S^A_{31}	S^B_{54}	S^B_{53}	S^B_{51}	S^B_{43}	S^B_{41}	S^B_{31}	S^C_{52}	S^C_{14}	S^C_{43}	Ri Ha Si
Overlap																									
A - 3,4			1	-1					1		-1			-1											4
A - 3,1	-1		1			-1	-1		1						-1										14
B - 5,4				-1	1											-1									5
B - 5,3			-1		1												-1								6
B - 5,1	-1				1													-1							9
B - 4,3			-1	1															-1						6
B - 4,1	-1			1																-1					9
B - 3,1	-1		1																		-1				9
C - 5,2		-1			1			-1					1									-1			4
C - 1,4	1			-1		1					-1	-1											-1		6
C - 4,3			-1	1					-1	-1	1	1												-1	8

Step (3). The longest job in the LP solution is job 1 and from this we get the objective:

$$\min\ t_f = 20 + t_{11} + S_{11} + S_{12}$$

A feasible solution with $t_f = 35$ is known for this problem and using this value it can be determined that $M = 40$ will satisfy (2b). This implies a group size of $(40)^{11}$.

6. Conclusion

When the group theory approach is applied to Job Shop Scheduling, the group problem can be easily obtained by inspection from the CPM solution of the associated LP problem. The group problem is characterized by considerable structure with all coefficient 0, 1, or -1.

Two problems arise when we apply this approach. First, the size of the group (M^k) is generally much larger than can be handled by present shortest route algorithms. Secondly, since we can construct at least one optimal solution to the group problem which violates the relaxed non-negativity constraints on basic variables, it appears that the relaxation of these constraints inherent in this approach will cause difficulty.

Future research in this area will address itself to these two problems. The special structure of the group problem will be studied to see if it can be exploited in developing an algorithm for solving the large group problems. In connection with the second problem, branch and bound approaches will be investigated for searching alternate solutions to the group problem to find one which satisfies basic non-negativity.[5]

---

[5]Shapiro, op. cit.

In this context the group problem could be used for getting lower bounds on partial solutions. Relaxations of the group problem which reduce its size and still produce a strong lower bound are being investigated. Finally, the use of Generalized Lagrange Multipliers in branch and bound procedures will be investigated.[6]

---

[6] Shapiro, J.F., "Generalized Lagrange Multipliers in Integer Programming," M.I.T. Working Paper 371-69, March 7, 1969, revised January 5, 1970.

REFERENCES

1. Conway, R.W., W.L. Maxwell, and L.W. Miller, Theory of Scheduling, Addison-Wesley, 1967, p. 103.

2. Fuchs, L., Abelian Groups, Pergamon Press, New York, 1960.

3. Gomory, R.E., "On the Relation Between Integer and Non-Integer Solutions to Linear Programs," Proceeding of the National Academy of Sciences, 53, pp. 250-256, (1965).

4. Gorry, G.A. and J. Shapiro, "An Adaptive Group Theoretic Algorithm for Integer Programming Problems," Technical Report No. 38, Operations Research Center, M.I.T., May, 1968. Revised May, 1969.

5. Herstein, I.N., Topics in Algebra, Blaisdell Publishing, Waltham, Massachusetts, 1964.

6. Hu, T.C., Integer Programming and Network Flows, Addison-Wesley, 1969.

7. Shapiro, J., "Dynamic Programming Algorithms for the Integer Programming Problem--I: The Integer Programming Problem Viewed as a Knapsack Type Problem," Operations Research, 16, pp. 103-121, (1968).

8. Shapiro, J., "Group Theoretic Algorithms for the Integer Programming Problem--II: Extension to a General Algorithm," Operations Research, 16, pp. 928-947, (1968).

9. Shapiro, J., "Turnpike Theorems for Integer Programming Problems," Sloan School of Management Working Paper 350-68, M.I.T., (1968). Revised August, 1969.

10. Wagner, H.M., "An Integer Linear-Programming Model for Machine Scheduling," NRLQ, 6, No. 2, June 1959.

Date Due

FEB 05 '78		
JUL 6 '76		
APR 02 '77		